

Comparing OLTP Scaling Behavior on Intel® Xeon™ and Itanium® 2 Processors

Richard Hankins^{o2}, Murali Annavaram^o, Brian Hirano¹, Jignesh Patel², John Shen^o

^oMicroarchitecture Research
Lab (MRL)
Intel® Corporation

¹Server Technologies
Oracle® Corporation

²EECS Department
The University of Michigan

Abstract

Analyzing On-Line Transaction Processing workloads can be quite challenging as small configurations may not accurately represent realistic systems, while large configurations are very complex to configure and cost prohibitive to build. This paper presents a comparative study of the scaling behavior of an OLTP workload on Intel® Xeon™ and Intel® Itanium® 2 processors. Using our “iron law” of database performance as a framework, we characterize the scaling behavior of two performance metrics that are critical to the transaction throughput: the average instructions executed per transaction (IPX) and the average cycles per instruction (CPI). This characterization is determined through an extensive empirical examination of an Oracle® based commercial OLTP workload running on both architectures. Our study shows that both systems’ CPI trend show two distinct regions of behavior as the workload size increases: a cached region and a scaled region. The intersection of the two regions, called the pivot point, is a minimal, representative workload configuration from which behaviors of much larger OLTP configurations can be accurately extrapolated. This study also provides an opportunity to verify our previously proposed hypothesis on how different system configurations will affect the pivot point, by showing that pivot point varies with system parameters in a predictable manner.

1 Introduction

On-Line Transaction Processing (OLTP) workloads are important benchmarks in the design of server processors. Analyzing OLTP workloads, however, can be quite challenging as realistic OLTP systems are complex to configure and can be cost prohibitive to build. Due to these constraints, researchers use very small, *cached* OLTP setups [2][3][5][7][9][15][16], which are cheaper and easier to configure. Researchers rely on the assumption that cached configurations have similar micro-architectural behavior as much larger, or *scaled* systems. Alternatively, some researchers [1][8][10][12][17] analyzed the behavior of OLTP workloads by monitoring scaled OLTP setups on

physical systems using embedded performance counters. These studies primarily focused on the analysis of the memory and disk I/O subsystems. Although these studies were based on data measurements from a physical system using a scaled setup, they are limited to one OLTP configuration. In particular, they did not focus on how the behavior varies with changing OLTP workload configurations.

In [4] we analyzed how an OLTP system behavior varies when the workload scales from a cached to a scaled setup. Our results showed that cached configurations can hide important system effects, such as frequent context switching and bus traffic overhead, that characterize scaled systems. Yet, our results also showed that both the Cycles per Instruction (CPI) and the Misses per Instruction (MPI) behavior follow predictable trends with workload scaling. These trends can be accurately approximated by two linear regions of behavior. We proposed a method for selecting the minimal OLTP configuration, called the *pivot point configuration*, which can be used as a basis for analyzing large-scale OLTP configurations.

Our observations in [4] were based on results generated from a single Quad Intel® Xeon™ SMP system. A question that remained unanswered in [4] is how the OLTP scaling behavior varies with different machine configurations. We now address that question by comparing the microarchitectural behavior of an OLTP workload executing on two different architectures: an Intel Xeon-based system and an Intel Itanium® 2-based system. We analyze the performance impact of the increased processor cache, memory capacity, and bus bandwidth of the Itanium 2-based system. In addition, we present results verifying our conjecture in [4] that the pivot point changes predictably with increased cache size and bus bandwidth.

The rest of the paper is as follows. In Section 2, we briefly describe our experimental framework. In Section 3, we use the “iron law” framework to show how CPI and IPX trends vary on both Xeon and Itanium 2 processors. In Section 4, we compare the behavior of the pivot points for both configurations, and show that the pivot point remains at a relatively small configuration. We conclude in Section 5.

2 Experimental Configuration

We will use the iron law of database performance [4] as the framework to understand how IPX and CPI vary over a wide range of processor and warehouse configurations on our two architectures. The OLTP workload used in this study is the Oracle® Database Benchmark, or ODB¹. A more detailed description of the iron law and ODB can be found in [4].

Table 1 shows the most relevant hardware and software configurations for the two systems. The two key differences to note are that (a) the Itanium 2 processor has a 3MB L3 cache as opposed to a 1MB L3 in the Xeon processor, and (b) the Itanium 2-based system is populated with 16GB main memory allowing us to use 14GB buffer pool, also called the System Global Area (SGA). Both processors provide a comprehensive set of embedded counters for monitoring performance in real time [6][13]. Performance monitoring on Intel processors is completely noninvasive and does not affect the execution of the Oracle software. To sample the performance-monitoring counters, we use EMON, a software tool used internally at Intel that is able to sample the counters at a user-determined frequency. A number of publicly available software products provide similar access to the performance-monitoring counters, including the Intel® VTune™ Performance Analyzer [14].

Component	Intel® Xeon™-Based System	Intel® Itanium® 2-Based System
Processors	4-way 1.6GHz	4-way 900MHz
Caches	256KB L2 1MB L3	256KB L2 3MB L3
OS	Red Hat® AS 2.1	Red Hat® AS 2.1
Disks	24 data + 2 log disks	32 data + 1 log disk
Memory	4GB	16GB
Database	Oracle® 9i Release 2 RDBMS	Oracle® Database 10g
OS Page Size	4MB	256MB
SGA	3GB	14GB

Table 1: System Descriptions

¹ ODB is not a compliant TPC-C Benchmark™, even though there may be similarities in the database schema and the transactions in the workload. Any results presented here should not be interpreted as or compared to any published TPC-C Benchmark results. TPC-C Benchmark is a trademark of Transaction Processing Performance Council (TPC).

2.1 Workload Scaling

In our experiments, we scale both the number of warehouses and the number of processors. For the Xeon-based system, we scale the number of warehouses from 10 to 1200. The Xeon-based system is I/O bound at configurations larger than 800 warehouses. For the Itanium 2-based system, we are able to scale the number of warehouses from 25 to 1500 warehouses, with configurations larger than 1200 warehouses becoming I/O bound. The increased number of disks (8 more disks), larger SGA and higher bus bandwidth on the Itanium 2-based system allow better workload scaling. For both architectures, we scale the number of processors from one to four.

In each configuration, we use a 15-20 minute warm-up period and a 10 minute measurement period. The warm-up period is determined empirically by observing the system’s transaction rate over time to verify that the transaction throughput is in a steady state after the warm-up. During the measurement period, a set of performance events are measured for 2 seconds in a round robin fashion, and measurements are repeated 6 times. The processor utilization is well over 90% during our measurement interval.

3 OLTP Trends on the Xeon-Based and Itanium 2-Based Systems

Following the framework defined by the iron law equation, we start with the overall performance as measured by the number of transactions executed per second, or TPS. We then focus on the IPX and CPI, the two most important components that determine throughput performance. We conclude this section by investigating the behavior of the L3 cache, and how this behavior contributes to CPI performance.

3.1 ODB Transactions per Second

Figure 1 and Figure 2 show the transactional throughput, in transactions per second (TPS), for both the Xeon-based and Itanium 2-based systems. As the number of warehouses increases, the Itanium 2-based system throughput decreases much more gracefully than the Xeon-based system throughput. The Itanium 2-based system has 16GB main memory of which 14GB is allocated to SGA for caching database buffer blocks. The larger SGA causes the disk I/O to increase at a slower rate with increasing number of warehouses. For instance, at a 100 warehouse configuration, the Xeon-based system reads approximately 4MB of data per second while the Itanium 2-based system reads slightly less than 3MB of data per second. In addition, the amount of data read per second by the Xeon-based system is

increasing at a rate of 7KB per warehouse, while the rate of data read per second by the Itanium 2-based system is increasing at a rate of 6KB per warehouse. As a result of the increased disk I/O activity, the Xeon-based system is spending twice as much time in the operating system as the Itanium 2-based system (approximately 20% for the Xeon-based system versus 10% for the Itanium 2-based system), which translates directly into additional latency to complete a transaction.

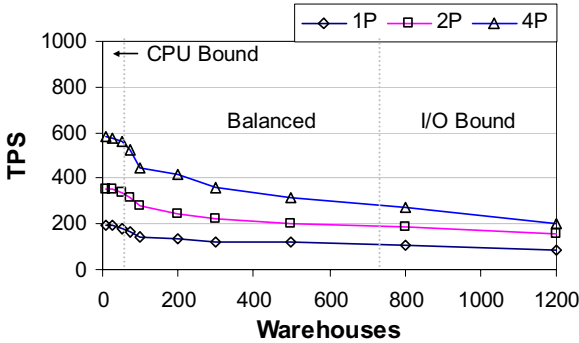


Figure 1: Throughput for Xeon-Based System

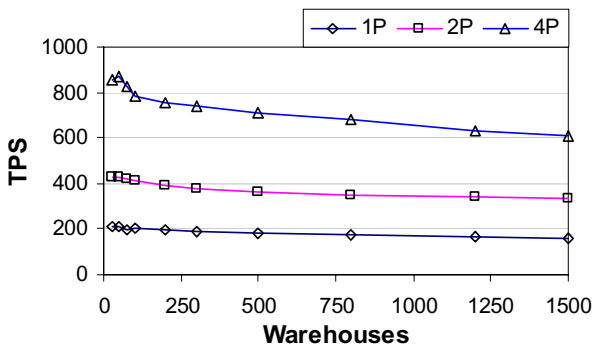


Figure 2: Throughput for Itanium 2-Based System

The throughput of the Itanium 2-based system scales almost linearly with the number of processors, while the throughput of the Xeon-based system can not achieve linear scale-up. From the iron law, the transactional throughput is determined by the instructions executed per transaction (IPX) as well as the CPI of the workload. As the number of processors scale, the front-side bus utilization increases. The Itanium 2-based system's bus bandwidth is twice that of the Xeon-based system, and because of the high bus utilization on the Xeon-based system (around 45%), the CPI increases with the number of processors, causing the performance to decrease. To

further investigate the reasons why the Itanium 2-based system scales better with number of processors, we now examine the IPX and CPI components of both systems.

For the experimental results that follow, we present data for configurations up to 800 warehouses on the Xeon-based system and for configurations up to 1200 warehouses for the Itanium 2-based system. Beyond these configurations, our systems become I/O bound and the processors are under utilized. The result is that a very tight "idle" loop of instructions will be executed continuously inside of the operating system, skewing the IPX and CPI statistics away from an accurate representation of the system.

3.2 CPI and IPX

Figure 3 and Figure 4 show the average number of instructions executed per transaction, or IPX, for the two systems. As the number of warehouses increases, the IPX of both systems also increases, with the Itanium 2-based system's IPX increasing at a much slower rate than that of the Xeon-based system.

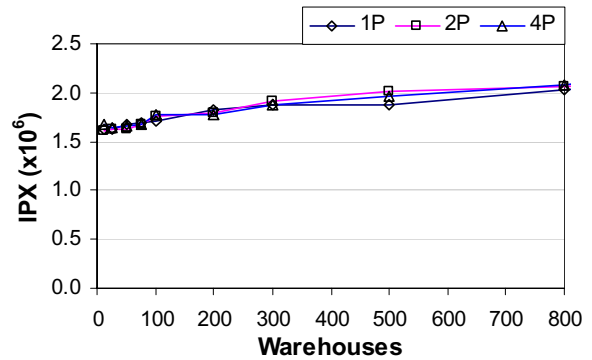


Figure 3: IPX for Xeon-Based System

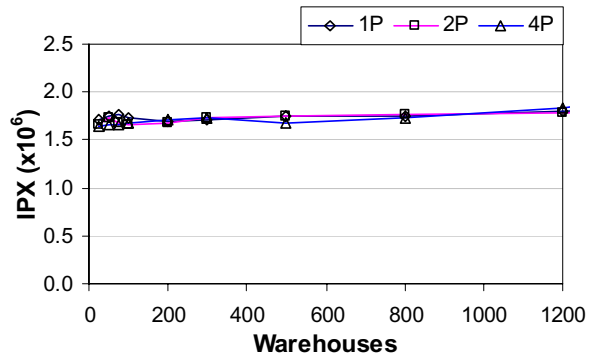


Figure 4: IPX for Itanium 2-Based System

IPX can be decomposed into two broad categories: instructions executed in the Oracle server (user space IPX) and instructions executed in the OS. Our results (not presented due to space constraints) show that user-space IPX remains nearly unchanged with increasing warehouses in both systems, indicating that the transaction code path through the Oracle server does not change with the number of warehouses. On the other hand, OS IPX increases proportional to the amount of I/O. The execution time in the OS is spent mostly servicing the I/O requests and context switching. As stated before, the I/O rate increases more slowly in the Itanium 2-based system, directly translating into a slower rate of increase in OS-space instructions executed per transaction. Furthermore, our results show that the ODB workload spends only 5-10% of its execution time in the OS on the Itanium 2-based system, compared with 10-20% of its time in the OS on the Xeon-based system. To explain the sub-linear throughput scaling with the number of processors, we now examine the CPI component of both systems.

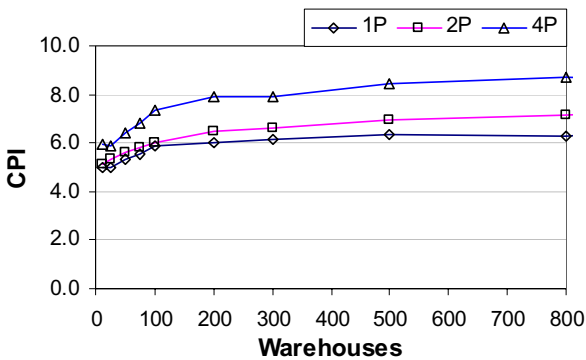


Figure 5: CPI for Xeon-Based System

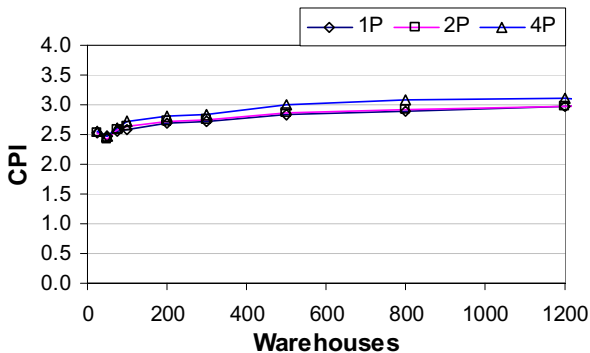


Figure 6: CPI for Itanium 2-Based System

Figure 5 and Figure 6 illustrate the average number of clock cycles per instruction executed, or CPI, for the Xeon-based and the Itanium 2-based systems, respectively. As the number of warehouses increases, the CPI of both systems increases as well. The most distinctive difference between the two systems is that the Xeon-based system's CPI increases significantly as the number of processors increases, while the Itanium 2-based system's CPI only marginally increases with the number of processors. In [4] we showed that the Xeon-based system's CPI increase can be attributed to the increased main memory access latency due to high bus utilization. The Itanium 2-based system has twice the bus bandwidth (6.4GB per second as opposed to 3.2GB in the Xeon-based system), and three times the L3 cache. The highest bus utilization approaches 45% for the Xeon-based system while the bus utilization for the Itanium 2-based system approaches 39%. Hence, the CPI increases only marginally with the number of processors on the Itanium 2-based system.

3.3 CPI Breakdown

To further understand the microarchitectural behavior of the workload, we decompose the CPI into its primary contributing components. For the Xeon processor, it is difficult to precisely determine how each microarchitectural event, such as cache misses, contributes to the overall stall time because the processor is able to overlap stalls through out-of-order execution. Hence, we use the simple approach of assigning a fixed number of CPU stall cycles to each microarchitectural event. The contribution of each event to the overall CPI is then calculated by multiplying the fixed stall cycles with the corresponding event count. The specific events that we have chosen as the primary contributing components to the CPI are the number of instructions executed, branch mis-predictions, TLB misses, Trace Cache (TC) misses, L2 cache read misses, and L3 cache read misses. After summing the resulting contributions, we arrive at a computed CPI. The difference between the measured CPI and the computed CPI is recorded as the "Other" component. Additional details on the methodology of this process, along with the estimated cost of each event, can be found in [4].

In contrast to the Xeon processor, the Itanium 2 processor executes instructions in-order and the performance counters can accurately decompose the execution time into six categories of microarchitectural events. The stall events are categorized as: Flush, RSE, L1d, FE, and EXE. The time spent retiring instructions is categorized as

Work. Table 2 provides a simple definition for each component and also provides events in the Xeon processor that approximately correspond to each Itanium 2 processor event category. It is important to note that the events listed as Xeon processor equivalents are very approximate and are only intended to illustrate the similarities. A more detailed description for each stall event in the Itanium 2 processor can be found in [11].

Itanium 2 Processor CPI Component	Description	Xeon Processor Equivalent (approximate)
Work	The time spent executing instructions	Inst
Flush	The stall time due to flushing the pipeline after a branch mis-prediction	Branch
RSE	The stalls due to the register stack engine	n/a
L1d	The stalls related to the L1 data cache and TLB	L1d + TLB
FE	The stalls in the back-end pipeline caused by stalls in the front end. Causes include I-cache miss stalls and TLB stalls.	TC
EXE	The stalls due to L2 cache, L3 cache, and main-memory accesses.	L2 Miss + L3 Miss

Table 2: CPI Component Definitions

Figure 7 shows the CPI breakdown for the Xeon processor. The components of the CPI are the instructions executed plus the processor stalls due to branch mis-predictions, TLB misses, trace cache (TC) misses, L2 read misses and L3 read misses. From the CPI breakdown it is obvious that the primary contributor to CPI is the L3 cache miss latency, which ranges from 60% to 75% of the overall CPI. Furthermore, the contribution of the L3 cache misses increases as the number of warehouses increases, while the contribution of all other events remain relatively constant. Note that it is difficult to split L2 and L3 misses into instruction and data misses on the Xeon processor. As a result L2 and L3 stall components account for both instruction and data cache bottlenecks.

Figure 8 shows the CPI breakdown for the Itanium 2 processor as the number of warehouses increases. The categories and their respective definitions are shown in Table 2. Unlike the Xeon processor, the Itanium 2 processor’s performance counters can accurately split the stall time due to instruction and data misses at L2 and L3. The FE component in each bar shows the stall time due to instruction misses, while the EXE component shows the stall time due to data misses. Hence, to compare the CPI contribution of cache misses shown in Figure 8 with those in Figure 7, one needs to compare the combined FE and EXE stalls in Figure 8 with the combined L2, L3 and TC stalls in Figure 7.

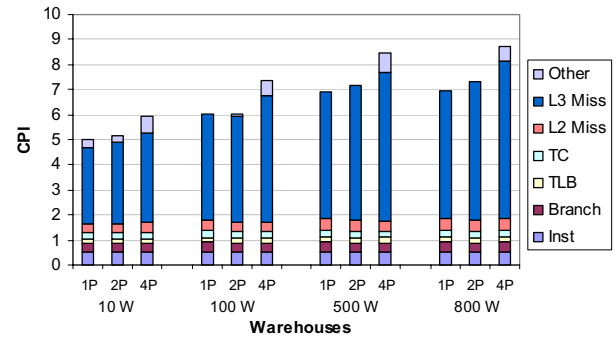


Figure 7: CPI breakdown for Xeon Processor

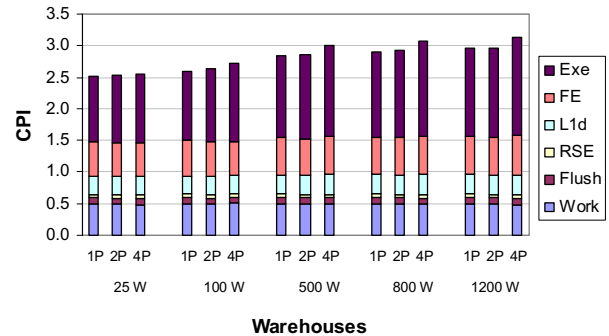


Figure 8: CPI breakdown for Itanium 2 Processor

Our results show that cache misses are the single largest contributor to the CPI in both the Itanium 2 and Xeon processors. Even with a 3MB L3 cache, the Itanium 2 processor suffers a significant number of instruction and data misses. The difference in CPI magnitudes between the two systems can be attributed to several differences in the configurations, main-memory latency, chipset, bus bandwidth, etc. For instance, the Xeon processor’s clock speed is 78% faster than the Itanium 2 processor’s, which

contributes to the higher access latencies in the memory hierarchy on the Xeon processor and, thus, the higher CPI. Similarly, the bus bandwidth on the Itanium 2-based system is almost twice that of the Xeon-based system which reduces the bus contention. Since L3 cache misses are the largest contributor to the CPI in both systems, we further explore the behavior of L3 cache misses by analyzing the cache miss trends.

3.4 MPI

Figure 9 and Figure 10 show the average number of L3 read misses per instruction executed, or MPI, for the Xeon-based and Itanium 2-based systems, respectively. For both systems, the MPI increases as the number of warehouses increases. Notice, the MPI does not increase as the number of processors scales, demonstrating that cache-coherence is not having an adverse affect on the number of cache misses.

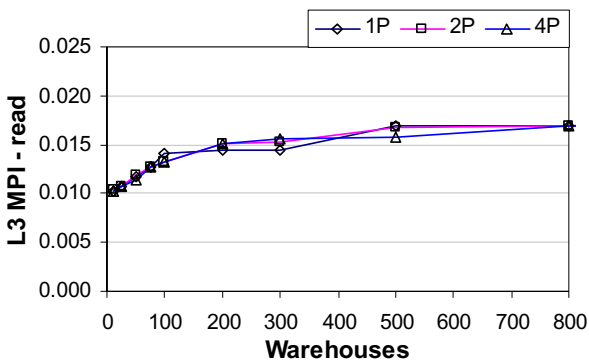


Figure 9: L3 MPI (read) for Xeon-Based System

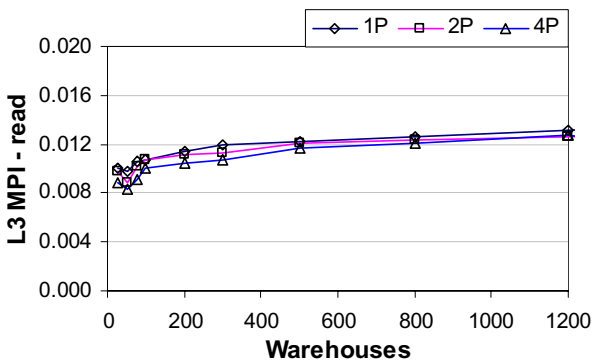


Figure 10: L3 MPI (read) for Itanium 2-Based System

The most striking difference between the MPI trends of the two systems is that the MPI for the Itanium 2-based system increases at a slower rate than the MPI of the Xeon-based system. The Itanium 2 processor has a 3MB L3 cache compared with the 1MB L3 cache on the Xeon processor. Furthermore, the higher rate of context switching in the Xeon-based system decreases the temporal locality of data accesses, increasing the overall cache miss rate. The Xeon-based system's SGA is much smaller than the Itanium 2-based system's SGA, which results in a higher I/O rate. A higher I/O rate leads to more blocking on reads and causes more context switches. The Xeon-based system executes 22 context switches per transaction at the 4P, 100 warehouse configuration, while the Itanium 2-based system executes less than 10 context switches per transaction for the same configuration. In addition, the Xeon-based system's context switches increase at a rate of 3 context switches per transaction for every 100 warehouses, which is an order of magnitude higher than the increase in context switch rate for the Itanium 2-based system.

4 Pivot Point

In our previous study [4], we showed that the micro-architectural (CPI) behavior of ODB can be approximated by two linear regions of behavior. In the cached region (small number of warehouses), the CPI increase is quite steep, reflecting the inability of the L3 to capture the working set. However, in the scaled region (large number of warehouses), L3 cache misses reach near saturation, and the CPI increase is quite gradual for increasing workload size. The pivot point is the workload size where the cached region's behavior ends and the scaled region's behavior begins. More importantly, the pivot point can be used as a lower bound to represent an OLTP workload with sufficient execution behavior to appear similar to a scaled setup. In our previous study, we also hypothesized how different system configurations will change the pivot point. Our current study provides us an opportunity to verify those hypotheses by comparing the pivot points on the Xeon-based and Itanium 2-based systems.

Figure 11 and Figure 12 show the pivot point for the Xeon-based and Itanium 2-based systems, respectively. Despite the vast differences in these two architectures, the trends in both figures are surprisingly similar. Architectures with larger processor caches and higher front-side bus bandwidth would exhibit lower MPI and lower cache miss latency. As a result, the slope of the CPI in the cached region would decrease, effectively pushing the pivot point to the right. At the same time,

increasing the size of SGA would allow more data to remain in the database’s buffer cache, reducing the I/O rate. A reduction in the I/O rate would decrease the MPI, as less context switching and less time spent in the OS would improve the cache hit rate. As a result, the slope and the magnitude of the CPI in the scaled region would decrease, effectively shifting the pivot point to the left. Thus, the L3 cache size, front-side bus bandwidth, SGA, and disk I/O bandwidth have opposing effects in determining the pivot point. The slopes for the cached and scaled regions of both systems are shown in Table 3.

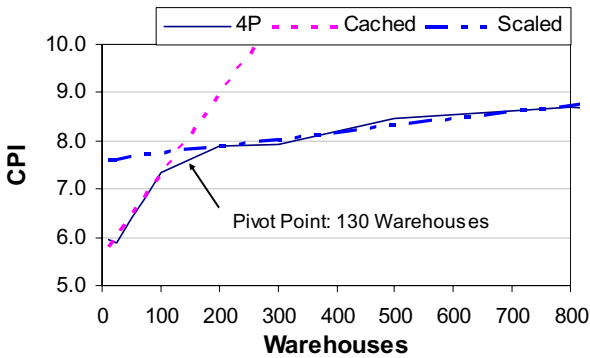


Figure 11: Pivot Point for Xeon-Based System

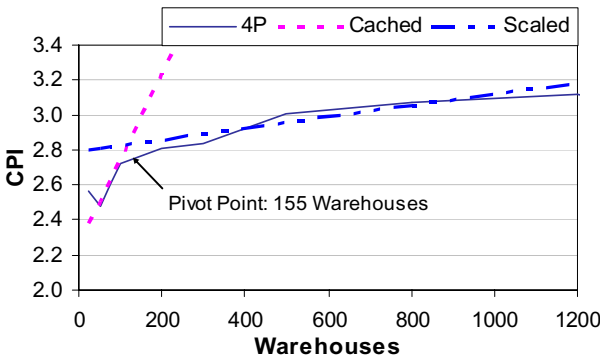


Figure 12: Pivot Point for Itanium 2-Based System

The resulting pivot point of 155 warehouses for the Itanium 2-based system is only slightly larger than the Xeon-based system’s pivot point of 130 warehouses, supporting our original hypothesis that the pivot point changes little despite the different architectures. As stated in our previous study [4], the pivot point is only a guide for one to use in determining the minimum workload size to use for study. Based on the results of this study, when using

architectures similar to the Xeon and Itanium 2 processors, one can study an ODB workload size of, say 200 warehouses, and then accurately extrapolate the resulting microarchitectural behavior to much larger configurations.

Region	Xeon-Based System (CPI per 1000 W)	Itanium 2-Based System (CPI per 1000 W)
Cached	16.67	2.508
Scaled	1.447	0.325

Table 3: Pivot Point - Slopes of the two regions

5 Conclusions

In our previous study [4], we examined the micro-architectural behavior of ODB, an OLTP workload, executing on a Xeon-based system. We have now completed the picture by analyzing the micro-architectural behavior for a second architecture, namely Itanium. Based on our new iron law of database performance, we show that both the IPX (average instructions per transaction) and CPI (average cycles per instruction) are critical to performance for both architectures. For database workloads running on small-scale multiprocessors, the primary impediment to processor performance is the penalty due to L3 cache misses. By increasing main memory capacity, incorporating larger L3 caches on die, and providing low latency high bus bandwidth, the OLTP performance will scale better with the number of warehouses and the number of processors.

Similar to the Xeon-based system, the Itanium 2-based system’s CPI trend shows two distinct regions of behavior as the workload increases: a cached region and a scaled region. The intersection of the two regions, called the pivot point, varies with system parameters in a predictable manner. The slope of the cached region is determined by the L3 cache size, which decreases with increasing L3 cache size. On the other hand, the slope of the scaled region is determined by the I/O rate, which decreases with increasing bus bandwidth. These two effects have offsetting influences on the behavior of the pivot point, with the pivot point configuration increasing modestly from 130 warehouses to 155 warehouses from the Xeon-based system to the Itanium 2-based system.

6 References

- [1] A. Ailamaki, D. DeWitt, M. Hill, and D. Wood. DBMSs on a Modern Processor: Where Does Time Go? In *Proceedings of the 25th*

- International Conference on Very Large Data Bases*, pages 266–277, September 1999.
- [2] A.R. Alameldeen and D.A. Wood. Variability in Architectural Simulations of Multi-threaded Workloads, In *Proceedings of the 9th International Symposium on High-Performance Computer Architecture*, pages 7-18, February 2003.
- [3] M. Annavaram, T. Diep and J.P. Shen. Branch Behavior of a Commercial OLTP Workload on Intel IA32 Processors. In *Proceedings of the International Conference on Computer Design*, pages 242-248, January 2001.
- [4] R. Hankins, T. Diep, M. Annavaram, B. Hirano, H. Eri, H. Nueckel, and J.P. Shen. Scaling and Characterizing Database Workloads: Bridging the Gap between Research and Practice. <http://www.microarch.org/micro36/html/pdf/annavaram-ScalingCharacDatabase.pdf>. To appear in the *Proceedings of the 36th Annual International Symposium on Microarchitecture*, December 2003.
- [5] L.A. Barroso, K. Gharachorloo, and E. Bugnion. Memory System Characterization of Commercial Workloads. In *Proceedings of the 25th International Symposium on Computer Architecture*, pages 3–14, June 1998.
- [6] Intel Itanium 2 Processor Reference Manual for Software Development and Optimization. . <http://www.intel.com/design/Itanium2/manuals/251110.htm>.
- [7] L.A. Barroso, K. Gharachorloo, A. Nowatzky, and B. Verghese. Impact of Chip-Level Integration on Performance of OLTP Workloads. In *Proceedings of the 6th International Symposium on High-Performance Computer Architecture*, pages 3-14, January 2000.
- [8] Z. Cvetanovic and D. Bhandarkar. Characterization of Alpha-Axp Performance using TP and SPEC Workloads. In *Proceedings of the 21st International Symposium on Computer Architecture*, pages 60–70, April 1994.
- [9] J. Lo, L. A. Barroso, S. Eggers, K. Gharachorloo, H. Levy, and S. Parekh. An Analysis of Database Workload Performance on Simultaneous Multithreaded Processors. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, pages 39-50, June 1998.
- [10] M. Franklin, W.P. Alexander, R. Jauhari, A.M.G. Maynard, B.R. Olszewski. Commercial Workload Performance in the IBM Power2 Risc System/6000 Processor. *IBM J. of Research and Development*, 38(5): 555–561, 1994.
- [11] S. Jarp. A Methodology for using the Itanium 2 Performance Counters for Bottleneck Analysis. http://www.gelato.org/pdf/Performance_counters_final.pdf.
- [12] K. Keeton, D.A. Patterson, Y.Q. He, R.C. Raphael, and W.E. Baker. Performance Characterization of a Quad Pentium Pro SMP Using OLTP Workloads. In *Proceedings of the 25th International Symposium on Computer Architecture*, pages 15–26, June 1998.
- [13] The IA-32 Intel® Architecture Software Developer’s Manual, Volume 3: System Programming Guide.
- [14] The Intel VTune Performance Analyzer. <http://www.intel.com/software/products/vtune/>.
- [15] P. Ranganathan and K. Gharachorloo and S.V. Adve and L.A. Barroso. Performance of Database Workloads on Shared-Memory Systems with Out-of-Order Processors. In *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 307–318, October 1998.
- [16] M. Rosenblum, E. Bugnion, S. Herrod, E. Witchel, and A. Gupta. The Impact of Architectural Trends on Operating System Performance. In *Proceedings of the 15th Symposium on Operating Systems Principles*, pages 285–298, December 1995.
- [17] K. Keeton, D.A. Patterson. The impact of Hardware and Software Configuration on Computer Architecture Performance Evaluation. In *the first Workshop on Computer Architecture Evaluation using Commercial Workloads*.